AFRL-IF-RS-TR-2004-280
Final Technical Report
October 2004

AGENTS OVERCOMING RESOURCE INDEPENDENT SCALING THREATS

Altarum

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. K275

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-280 has been reviewed and is approved for publication

APPROVED: /s/

Richard A. Hyle
Project Engineer

FOR THE DIRECTOR: /s/

JAMES A. COLLINS, Acting Chief
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
OCTOBER 2004

3. REPORT TYPE AND DATES COVERED
Final Aug 02 – Mar 04

4. TITLE AND SUBTITLE
AGENTS OVERCOMING RESOURCE INDEPENDENT SCALING THREATS

5. FUNDING NUMBERS
C - F30602-00-C-0134
PE - 62301E
PR - ANTS
TA - 00
WU - 01

6. AUTHOR(S)
H. Van Dyke Parunak, Sven Brueckner,
John A. Sauter, and Robert Savit

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Altarum
PO Box 134001
Ann Arbor Michigan 48113-4001

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IF
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2004-280

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Richard A. Hyle/IFTB/(315) 330-4857/ Richard.Hyle@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*
This project uses abstract simulation models of resource allocation and mathematical techniques inspired by statistical physics to study the nonlinear emergent dynamics of distributed decentralized resource allocation. Our techniques seek to characterize the dynamics that may be anticipated in real systems, to predict pathological dynamics such as peaks in required computational effort and catastrophic breakdown in performance, and to develop control methods based on this understanding.
Our general approach begins with a set of abstract Resource Allocation Games (RAG). These games are derived from the Minority Game, a simple model of competition for scarce re-sources that captures essential features of interactions among agents that are heterogeneous, autonomous, boundedly rational, adaptable, parallel, co-situated, and experi-enced. Our research explores and generally confirms two hypotheses concerning the dynamics of resource allocation.

- The Generality Hypothesis asserts that a generic RAG exhibits dynamics that are intrinsic to resource allocation, independent of mechanism.
- The Specificity Hypothesis asserts that a RAG can be developed to resemble a specific re-source allocation mechanism and study its (idiosyncratic) dynamics.

14. SUBJECT TERMS
Fine-Grained Agents, Dynamics, Negotiation, Stigmergy, Pheromones, Statistical Physics, Minority Game

15. NUMBER OF PAGES
20

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

1 Summary

This document summarizes activities and results in the AORIST project under the DARPA IXO Autonomous Negotiating Team (ANT) program subsequent to those reported in the Interim Report of July 2002. Together with that report, it constitutes the final report of the project.

Altarum's activities during this period have been devoted to two lines of programmatic support and two lines of scientific investigation.

Programmatically, Altarum contributed a solver based on our pheromone learning mechanisms to the 3 June 2003 demo of the Schedules Negotiated by ANT-based Planner/ Maintenance PLANing Tool (SNAP/MAPLANT) system, and coordinated production of the ANT eBook. The Altarum solver (called RAPSIDy, "Resource Allocation Problem Solver Incorporating Dynamics") is a repair solver: it starts with an existing solution and modifies it incrementally, thus minimizing the impact of rescheduling on existing commitments.

Altarum's scientific work during this period has focused on two topics. First, we have explored the feasibility of a quick review tool (QuiRT) that would use rapid characterizations of a task allocation matrix to estimate the solvability of the matrix. Second, we have extended the resource allocation game (our generalization of the minority game) to arbitrary networks of interacting resource allocators, thus enabling exploration of the dynamics of logistical systems such as supply networks Logistics Resource Allocation Game (LogRAG).

This work has led to new take-aways in addition to those described in our interim report.

- We have begun developing a metaphor based on the statistical mechanics concept of universality to characterize multi-agent system and understand when refinements to agent reasoning may or may not pay off in improved system performance.
- The LogRAG framework promises to be a useful modeling tool for leaders in the DoD logistics community.

We are actively pursuing transition of these tools and techniques in several areas:

- Supply network engineering for defense contractors;
- Personnel management for the Navy;
- Modeling dismounted infantry;
- Military logistics.

In addition, we have produced several new publications covering work done during the entire period of the project.

2 ANT Program Support

Altarum contributed a solver based on our pheromone learning mechanisms to the 3 June demo of the SNAP/MAPLANT system, and coordinated production of the ANT eBook.

2.1 RAPSIDy Solver for ANT Demo

RAPSIDy (Resource Allocation Problem Solver Incorporating Dynamics) is a repair solver based on some of the concepts developed by the Altarum team during the ANT program. By "repair solver" we mean one that begins with an existing schedule and a set of new constraints and attempts to deliver a scheduled that incorporates the new constraints with minimal change to the previous schedule.

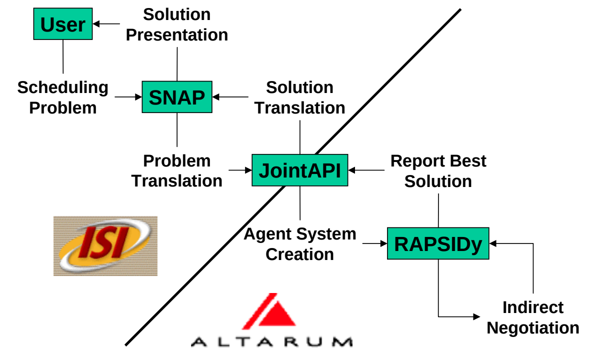

Figure 1. RAPSIDy directly interfaces with ISI/SNAP to solve (re-)scheduling problems.

RAPSIDy interacts with the ISI SNAP system (Figure 1). It receives the definition of the current scheduling problem as well as user preferences and requirements and it delivers a set of assignments that (partially) solve the problem under the given constraints.

A scheduling problem as presented by SNAP (Figure 2) comprises a set of resources (pilots, planes, ranges, simulators, …) and a set of tasks (missions). Tasks specify a number of requirements that all have to be met to fulfill the task. Requirements provide the constraints on the eligibility of resources to be assigned to this task. A task is divided into contiguous temporal segments, which jointly define the overall duration of the task. A requirement of the task may be associated with one or more of these segments and thus, a resource may be assigned to a task (through the specific requirement) for only parts of the overall task duration.

Many of the constraints on the allocation of resources to the requirements of a task at a given interval within the overall planning horizon are hidden from the operation of a solver (such as RAPSIDy) connected to SNAP. Instead, a solver explores the permissibility and value of potential task assignments through interactions with a domain Oracle inside the SNAP system. This arrangement has the advantage of avoiding the explicit representation of constraints arising in a specific domain within the solver. Instead, all the solver knows about is a general scheduling problem (assign resources to requirements at specific times) with the potential for failure of task configurations that are prohibited by the Oracle even though they meet the requirements of the general problem structure. On the other hand, the lack of the explicit representation of constraints leads to a significant overhead in the operation of the solver, since it needs to explore regions in the search space of the general problem even though they are excluded by the specific domain constraints. This overhead led to a significant slow-down of the solver operation, especially in the region of problem space, where the abstract capacity of the system (availability pattern of resources) is near the abstract demand by the tasks (requirement pattern). In this case, the resulting pattern of permissible and forbidden configurations in search space is very complex and the

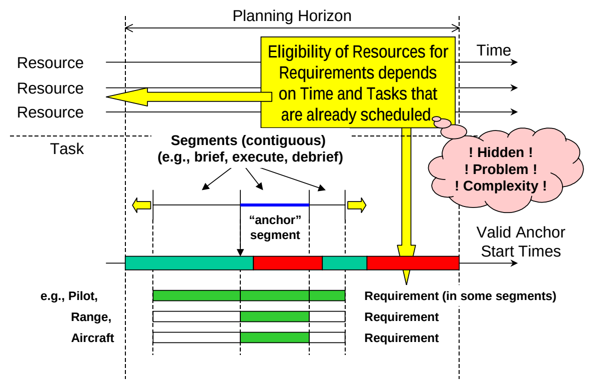

Figure 2. A scheduling problem comprises resources, tasks and their associated requirements.

Oracle needs longest in deciding whether a configuration is acceptable or not (peak in effort curve).

RAPSIDy implements a local distributed hill-climbing approach (Figure 3). Each task in the problem definition is assigned an agent. The agent's goal is to find an acceptable assignment for all its requirements within the space of resources and the planning horizon interval. This Resource-Timeline Environment (RTE) provides the shared environment of the task agents, in which they may experience the domain-independent scheduling constraints (e.g., tasks may not overlap) without consulting the Oracle. For all domain-specific constraints (e.g., eligibility of a resources for a requirement at a specific time), task agents interact with the Oracle.

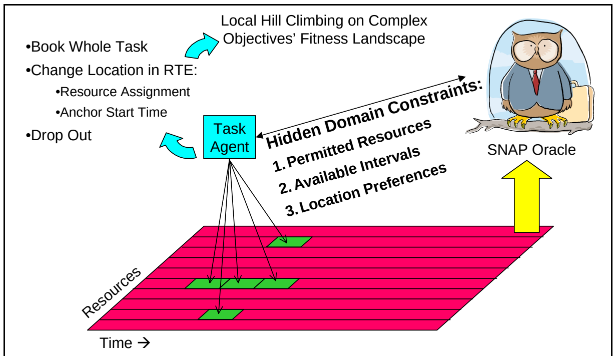

Figure 3. RAPSIDy performs a local distributed hill-climbing search for the best global assignment.

Once the task agent has found an acceptable assignment in the RTE, it then explores local variations of this assignment to improve its evaluation by the Oracle with respect to the user preferences (Figure 4). Local variations of the assignment include changes in the temporal allocation of the task (slide up or down the timeline) or in the individual assignment of resources to specific requirements. At this point, RAPSIDy only considers user preferences with respect to a previously constructed schedule. Tasks seek to change their assignment towards configurations that match this "old" schedule. Thus, the RAPSIDy solver attempts to repair an existing schedule that needs to be modified either because new tasks have been added to the problem, or the resource availability has changed in some way. It is the goal of RAPSIDy to create a new schedule that resembles the old schedule as much as possible and thus minimizes the changeover costs involved with distributing the new schedule information.

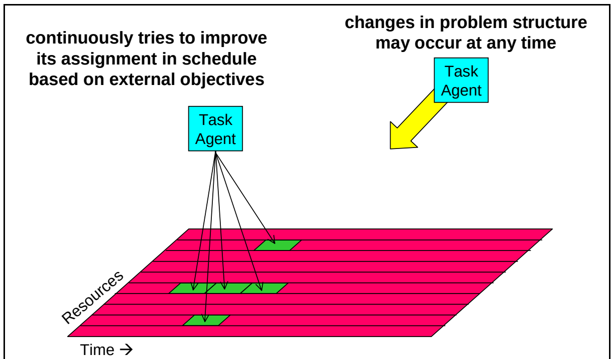

Figure 4. Task agents enter the RTE with the permission of the Oracle and then explore local variations of their assignment to improve the overall solution.

We were able to compare the RAPSIDy solver with two other scheduling approaches constructed by other ANTS teams. Compared to the SerialCrawler, a greedy assignment approach with systematic backtracking, RAPSIDy performs slower but delivers better solutions especially for more complex problems. Alternatively, when compared to the PseudoBoolean solver, RAPSIDy's lack of detailed temporal reasoning reduces the quality of solutions to problems of extreme complexity. In such problems, the space of acceptable solutions is reduced to single points rather than regions that could be explored by our solver. But RAPSIDy finds slightly less valuable solutions in much shorter time.

RAPSIDy was integrated with the ISI SNAP system and demonstrated at the 3 June 2003 scheduling workshop.

2.2 eBook

Altarum chaired a group of contractors who compiled an ANT eBook, an electronic snapshot of the research developed during the ANT program [3] This compilation includes detailed descriptions of the two challenge problems addressed by the ANT program, a survey of the modeling tools and techniques developed by ANT contractors (including demos of some of the more mature tools), and a catalog of reusable software with points of contact.

3 QuiRT

The objective of QuiRT, the Quick Review Tool, is to identify heuristics that can quickly review a resource allocation problem of the Marbles form to provide a quick estimate of the difficulty of solving it and bounds on the value of potential solutions. Such a tool would be invaluable in making more effective use of the more detailed solvers embedded in CAMERA(Coordination and Management Environments for Responsive Agents)-MAPLANT, since full solution of a problem by these systems takes more than an hour, too long for interactive refinement. In our interim report, section 4.3.2, we describe some initial work toward this objective, including the VPR (value per resource) estimate that gives a close upper bound of the recoverable value. In this period, we did further work on estimating the difficulty of solution. Our results are for the most part negative, demonstrating that some promising ideas do not in fact discriminate easy from difficult problems. In this section we summarize the form that QuiRT might take, review the VPR metric, and survey our work on estimating difficulty.

3.1 Notional QuiRT Interface and ConOps

Figure 5 and Figure 6 show a notional QuiRT interface that captures the basic concept of operation that QuiRT technology would support. The screen has four areas.

- The upper left corner of the screen shows a problem in the form of a resource eligibility matrix. For clarity, in this sketch the matrix does not distinguish separate requirements within tasks.
- The upper right corner plots the maximum retrievable value estimated for this problem (vertical axis) against the feasibility of solution (inverse of difficulty, horizontal axis). The upper right-hand corner of this plot is the most desirable location for a problem, indicating that it can easily be solved and will yield high value. The lower-left corner is the least desirable location, occupied by difficult problems likely to yield only little payoff.
- The bar charts at the bottom of the display resources ordered by a measure of the demand or load that they face (left) and tasks ordered by an estimate of how constrained

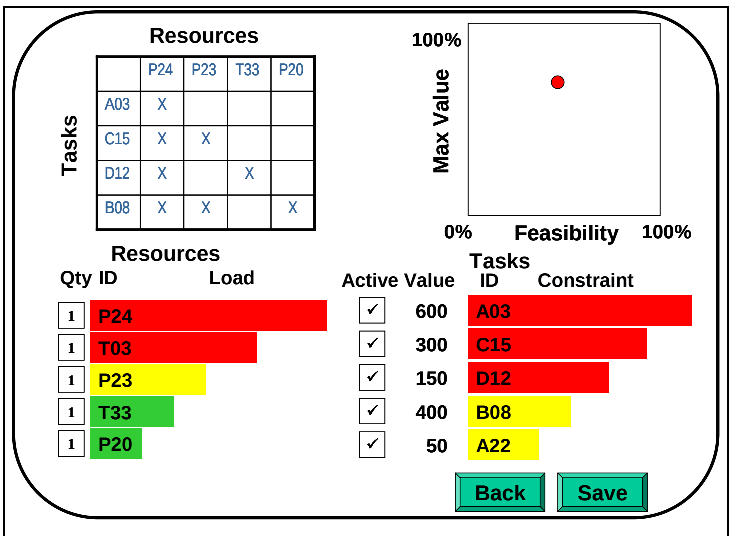

Figure 5: Notional QuiRT Interface (Step 1)

they are by resource availability (right). These bars may be color coded to indicate levels of load or constraint that are unacceptably high.

This display provides a planner with the information needed to modify the problem before submitting it for solution. Modifications can take several forms, including adding or removing eligibility indications in the problem matrix, changing the value associated with individual tasks, changing the quantity of each class of resource that is available, or changing which tasks are active in the problem.

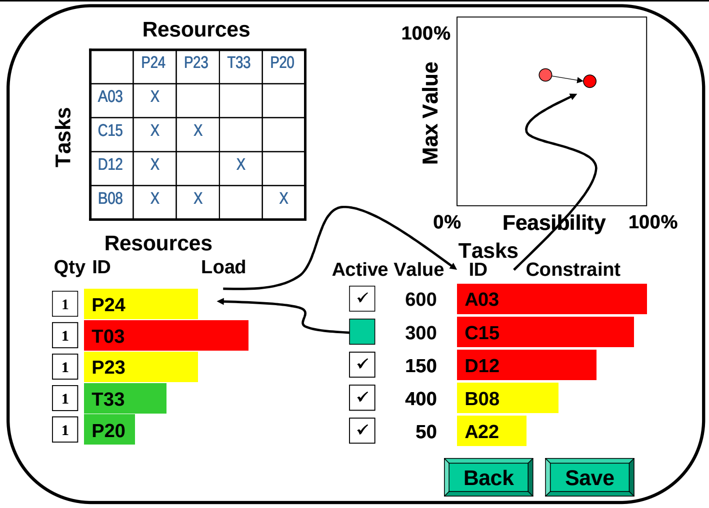

Figure 6: Notional QuiRT Interface (Step 2)

In this example, a planner might notice that resources P23 and P24 are highly overloaded, and that task C15 is attempting to get both of them. Perhaps removing C15 from the problem would improve the problem's location in the space of value x feasibility. Figure 6 shows a possible result after such a change. Deselecting C15 drops the load on resources P23 and P24, lowering the constraint on Task A03. The result is to make the problem more feasible while only slightly reducing the maximum recoverable value.

Such a vision requires a quick way to estimate the location of a problem in the value x feasibility space, and the challenge of QuiRT is to develop such estimators.

3.2 Estimating Value: the VPR Metric

A fairly effective upper bound of the value recoverable from a problem can be calculated as follows:

1. Calculate the a value-per-resource required (VPR) for each task by dividing the task value by its number of requirements (and therefore needed resources).
2. Rank all tasks by their VPR from high-to-low.
3. Select tasks for inclusion in the solution upper bound set by choosing the highest VPR tasks until they have used all the available resources. If the last task can only be partially satisfied by the remaining resources, include a pro-rated portion of the task proportional to the available resources.
4. Add up the task values in the upper bound set, and this becomes the upper bound value of a solution.

This estimate is an upper bound because no consideration is given to resolving resource conflicts, and this solution may not be feasible (and usually isn't). Nevertheless, it is a useful metric for evaluating experimental runs, and has proven to be quite tight. Figure 7 compares the upper-bound to the Marbles algorithm and also to the Contention Estimate described in the interim report.

3.3 Estimating Difficulty

Estimating problem difficulty has proven to be a much more challenging task. We had very limited resources to devote to this question, and have been able only to establish some preliminary and largely negative results. This section describes how we generated experimental problems, and experiments with several algorithms: for solving or classifying them.

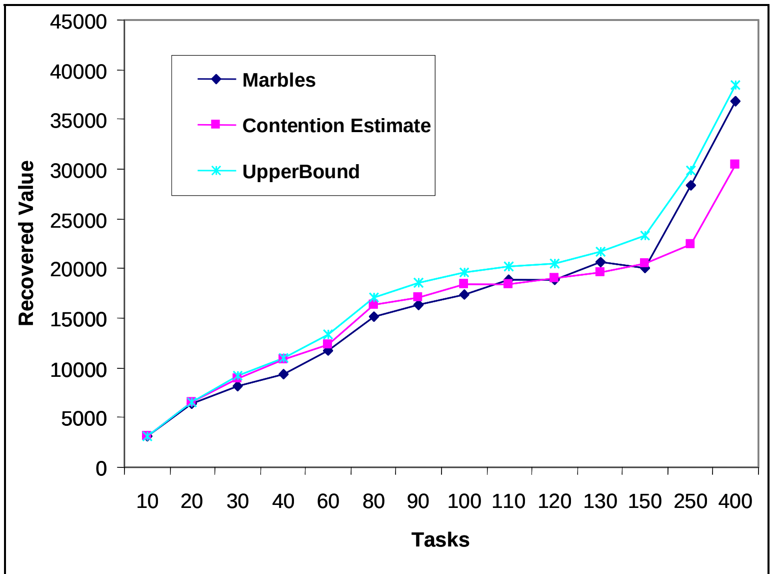

Figure 7: Comparison of (VPR) Upper Bound with Marbles performance and Contention Estimate

3.3.1 Generating Experimental Problems

In this preliminary stage of the investigation, we have focused our attention on Marbles problems with only one requirement per task. This problem, which is known in the Operational Research (OR) literature as the Assignment Problem (AP), is tractable. The Hungarian Algorithm, known since 1955, can solve it or show that no solution exists in O(N^3) [6], and more efficient methods have since been discovered. Our motive in beginning with such a simple problem is our need to develop intuitions about how the degree of a problem's difficulty manifests itself. Initial experiments require readily computed solutions as baseline comparisons. Even within the AP we see a range of problem difficulty, and if a metric cannot discriminate these, it is not likely to be useful when applied to intractable problems.

In generating test problems, we focus on square matrices with *N* rows and columns and *k* >= *N* marked cells. (Any matrix with more tasks than resources is unsolvable, and any matrix with more resources than tasks can be reduced to a square matrix whose solution also solves the initial problem.) We define a Trivially Unsolvable Matrix (TUM) as a matrix in which at least one row or column is empty. We considered three mechanisms for generating test matrices.

Seeded Matrices.—The simplest approach to generating solvable matrices is to begin by filling the diagonal, then randomly assign the remaining *k* – *N* marks to other cells. However, we want to generate a problem set that includes unsolvable matrices as well, and all seeded matrices are solvable by construction. (Figure 8 proves that there are unsolvable matrices that are not TUM's.)

Nontrivial Matrices.—At the other extreme, we can assign *k* marks randomly to the cells of the matrix, and afterward discard any matrices that turn out to be TUM's. This approach provides the most neutral distribution of test cases. However, when *N* is large and *k* ~ *N*, virtually every matrix generated is a TUM, and generating a useful set of test cases is extremely time consuming.

X		
X		
	X	X

Figure 8: An Unsolvable Matrix that is not a TUM

Modified Nontrivial Matrices.—To get reasonable yield of experimental cases, we follow the Nontrivial method until the number of marks remaining to be assigned is equal to the maximum of the number of unmarked rows or the number of unmarked columns. Then the range of cells to which the remaining cells are assigned randomly is restricted to cells in an unoccupied row or column.

3.3.2 Algorithms

This section describes several algorithms we tested on these problems. The first three try to generate solutions. In the first two, failure to generate a solution does not prove that no solution exists, but in the third it does. The fourth algorithm attempts to detect difficult problems without constructing a solution. All of these mechanisms, unlike the Hungarian Algorithm, can be applied to problems more complex than the AP.

3.3.2.1 Greedy Solver

Perhaps the simplest approach is to

- select a marked cell at random
- make the assignment that it indicates,
- remove all other markings from the marked cell's row and column,
- repeat until problem is solved or no markings remain.

This extremely simple algorithm serves as a benchmark for other approaches. Figure 10 shows how the percent of matrices solved by the Greedy method varies as a function of *k* for seeded and nontrivial test cases. These results lead to two observations.

First, we observe the easy-hard-easy profile that we have seen in other contexts [8], reminding us that adding flexibility (in this case, more marks) to a highly-constrained system can increase the difficulty of solution.

Second, observe that making a single assignment moves our location on these plots in two directions. First, it reduces the *N* of the remaining problem by one, thus moving vertically to the curve for the next lower *N* and increasing the percentage of success. Second, it reduces *k* by the number of marks in the selected row and column, moving us to the left. In the region to the right of the mode in these plots, where percentage solved is increasing with *k*, this shift has the effect of reducing the percentage solved.

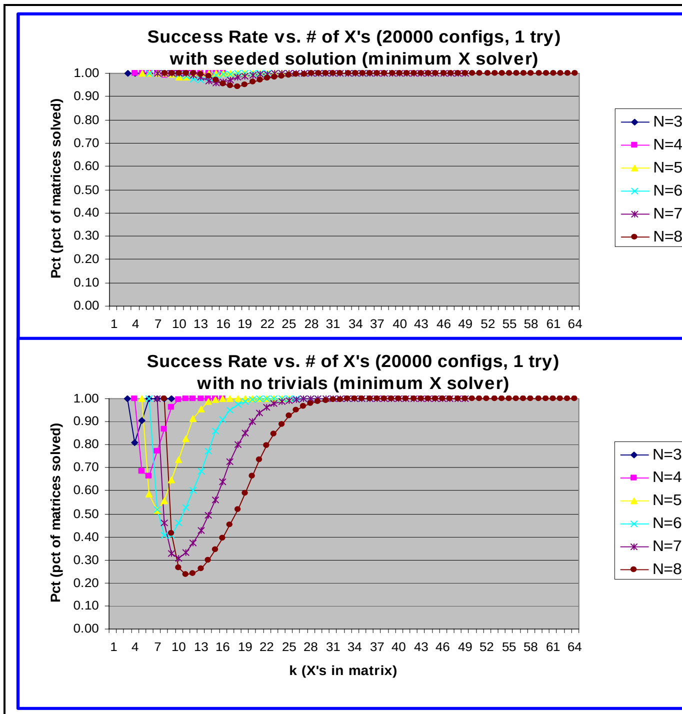

Figure 9: Percentage Solved with Min-X solver.— Top: seeded matrices. Bottom: nontrivial matrices.

3.3.2.2 Min-X Solver

The second observation on the results from the greedy solver suggests a heuristic of selecting the assignment that reduces *k* the least. We call this the Min-X strategy. Figure 9 shows the results. This heuristic significantly outperforms the greedy approach, but still fails to find about 5% of the solvable problems cases in the most difficult region on a single try.

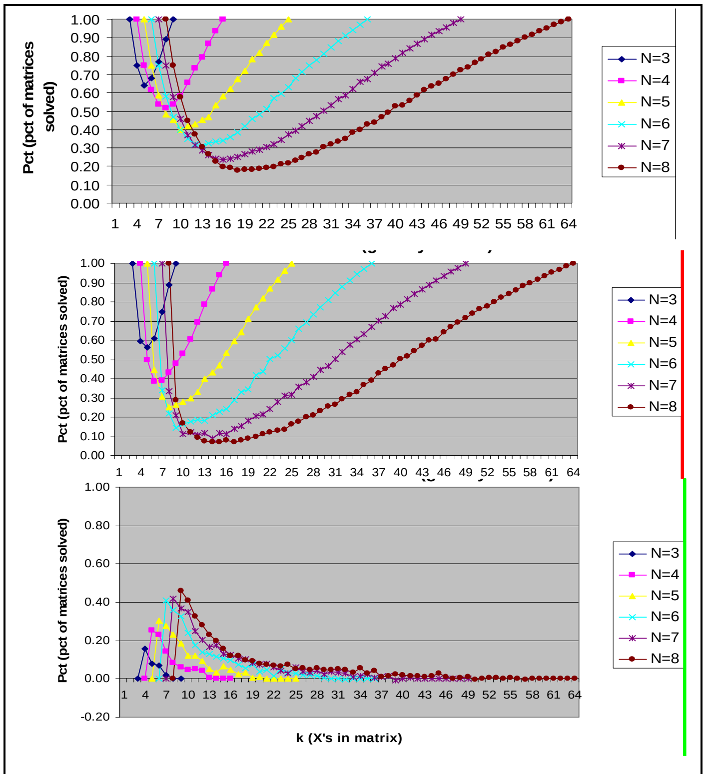

Figure 10: Percent of test matrices solved by Greedy solver. Top: seeded data. Middle: nontrivial data. Bottom: difference between top two plots.

3.3.2.3 Sauter's Sieve

The intuition behind the TUM can be extended to define high-order versions of the TUM. We use the term "filade" as a generic term to describe either a row or a column. If we are discussing one filade, a "counterfilade" is another filade orthogonal to the one under discussion. A TUM has one filade with no marks in its intersection with any counterfilade (Figure 12). Similarly, a matrix with two filades having marks that intersect with at most one counterfilade is unsolvable (Figure 11). In general, any matrix that has *n* filades whose intersections with fewer than *n* counterfilades are unmarked is unsolvable at level *n* – 1. (Thus a TUM is unsolvable at level 0.)

This notion can be used to define another approach to solving problems, named "Sauter's Sieve" after its inventor. Define an "obligatory kernel" as a set of *n* filades with *n* marked intersections with counterfilades. Then repeatedly

- select a smallest obligatory kernel
- make assignments
- repeat until either the problem is solved, or the problem has been reduced to a Level-*m* unsolvable matrix.

It is an open question whether Sauter's Sieve is complete (whether it will in fact find all unsolvable problems). We expect that a completeness proof can be constructed from Hall's Marriage Theorem [5].

X			
X	X		
	X	X	X

Figure 12: A Level-0 Unsolvable Matrix.— One filade (row 3) has fewer than one marked intersections.

X			
X	X		
	X		
	X	X	X

Figure 11: A Level-1 Unsolvable Matrix.— Two filades (columns 3 and 4) have fewer than two marked intersections.

3.3.2.4 Entropy Measures

The definition of multiple levels of unsolvableness that underlies Sauter's Sieve leads to another intuition. A problem in which marks are clustered in a few filades will be harder to solve than one in which they are distributed more evenly across the filades. This leads us to hypothesize that high filade entropy should correlate with low solvability. Such a measure would have computational complexity linear in the size of the matrix, and so would be much more efficient than measures (such as Sauter's Sieve) that require actually attempting to solve the matrix.

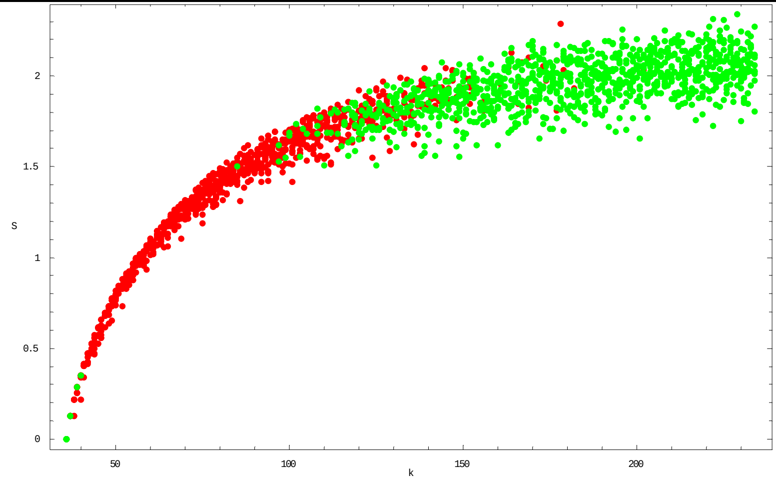

Figure 13: Row Entropy of *N* = 36 matrices as a function of *k*.—Red dots indicate unsolved matrices; green dots indicate solved matrices (solvability based on ten trials with Min-X solver).

To test this hypothesis, we generated a set of modified nontrivial problems with $N = 36$. We define filade entropy $F = - \sum p_i \log(p_i)$, where p_i = (population of ith filade) / k. We estimate a problem's solvability by running the Min-X solver ten times, and declaring the problem unsolvable if no solution is found in ten trials.

Figure 13 plots the row entropies as a function of k, marking solved matrices with a green dot and unsolved ones with a red dot. Expected features include:

- Entropy increases with k (since larger values of k provide more combinatoric options for row occupation numbers)
- Unsolved matrices occur for intermediate values of k, while low and high values of k yield solvable matrices (the "easy-hard-easy" pattern discussed above).

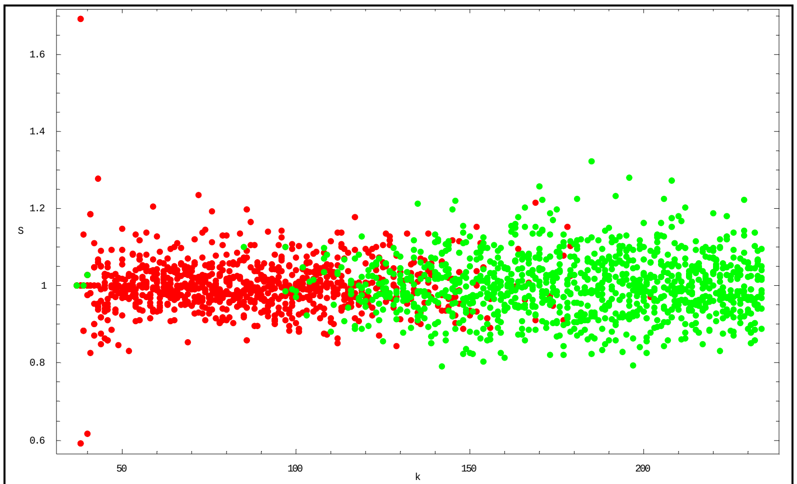

Figure 14: Ratio of Row Entropy to Column Entropy.—Unsolved matrices are red, solved ones are green.

What we are seeking in this plot is evidence that the entropy differs systematically as a function of solvability. There is no evidence of such a dependency.

Because of symmetry, there should be no systematic difference between row and column entropies. Figure 14 checks this by plotting the ratio of row entropy over column entropy for each matrix. As expected, the distribution is flat.

The result of this exploration is that filade entropy, applied in the simplistic manner discussed here, does not estimate the solvability of a matrix.

4 Logistics RAG

In the first period of the project, we generalized the Minority Game (MG [4]) in a number of ways to make it more directly comparable to resource allocation problems [10]. The resulting resource allocation game (RAG, and a simplified version known as the MiniRAG) permits independent variation of supply and demand, supports more than two suppliers, and allows several forms of partial awards. Still, the (Mini)RAG (like the MG) describes only the allocation of one commodity among a population of suppliers and consumers. To apply our methods to military logistics systems, we have begun to develop ways to cascade several MiniRAG's together to form a logistics Resource Allocation Game (LogRAG). This section describes the structure of the LogRAG and some preliminary experimental results. Full exploration of the dynamics of this coupled resource allocation model await future programs.

4.1 Structure of the Logistics RAG

Our MiniRAG model has three main components. It comprises a population of **Consumer** agents that repeatedly choose among members of the **Supplier** agent population to request one **Resource** each. Figure 15 shows a particularly interesting configuration of the MiniRAG model, in which there is one more consumer than there are resources at the two suppliers. We call this configuration the Minority Game configuration, since its competition dynamics are those of the widely studied Minority Game.

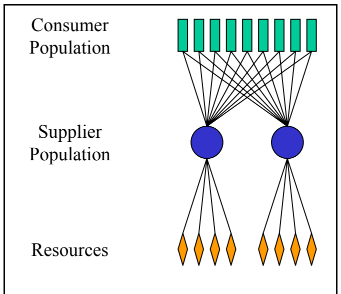

Figure 15. MiniRAG components form one competition process.

The LogRAG model introduces another component, the Site, which structures our populations in a network, but does not add any new dynamic process. In our model, we assign each supplier its own site. The supplier has its resources, and we require that each resource is associated with at least one consumer at that site, which provides the "input material" to make the resource. The number of consumers per resource is fixed for each site and all consumers of a resource have to succeed in their bidding before the resource becomes actually available to the supplier. Our model assumes that the consumer continues bidding until it succeeds and then it holds the input material until its peers have all acquired their input to collectively make the resource.

Figure 16 shows an example of a LogRAG configuration with a network of 6 sites. The network structure determines which suppliers are accessible to the consumers of a particular site to send bids for their input material. In the example we see that the competition for resources is more complex than in the MiniRAG model, since consumers compete not only with other consumers at the same site (as between the top and middle layers), but also with a dynamically changing subset of consumers from neighbor-

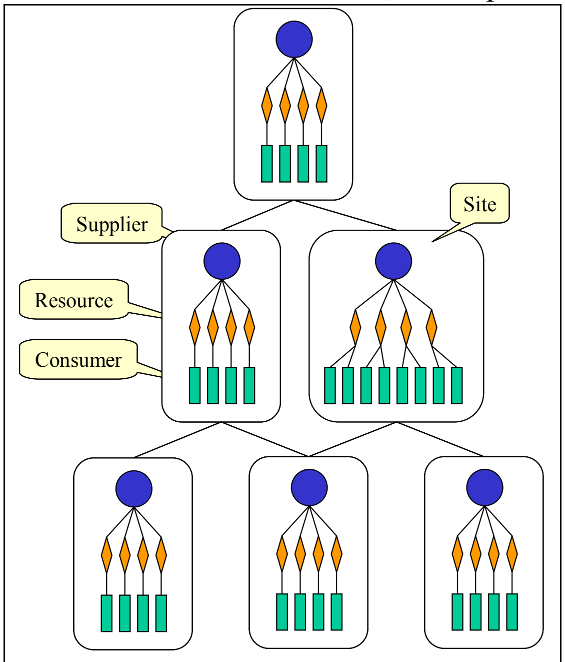

Figure 16. Resource competitions interacting in a network.

ing sites (as between the bottom two layers).

Some of the sites in the LogRAG network are special. First, there are sites at which the consumers do not have access to any suppliers (the bottom layer in Figure 16). These are considered "Raw Material Suppliers (RMS)" in the supply network, which are therefore assumed to always have the full set of resources available for distribution (material flow into the network). RMS resources do not require any consumer agents. Secondly, there are sites to which no consumer has access (the top layer in Figure 16). These are considered the "Original Equipment manufacturer (OEM)" of the supply network. Resources that are produced at an OEM site are immediately consumed (material flow out of the network).

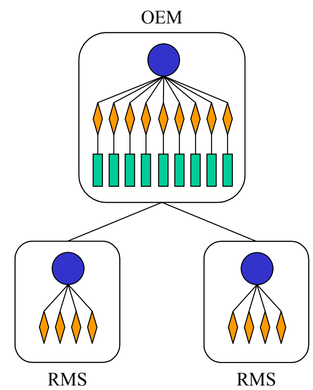

Figure 17. LogRAG in MiniRAG Minority Game configuration.

4.2 Initial Experiments

We implemented the LogRAG model using our in-house agent simulation framework and we integrated the model into our parameter sweep infrastructure to perform systematic exploration of the model's parameter space.

To test the code of our implementation, we configured the LogRAG to match the MiniRAG. Figure 17 shows the LogRAG configured to reproduce the Minority Game dynamics of the MiniRAG. The configuration comprises one OEM and two RMS sites, there is one more consumer at the OEM site than there are resources at the RMS sites combined, and the resources at the OEM site only require one consumer to provide input material.

The plots in Figure 18 show the outcome of a series of experiments in which we kept the capacity of the two RMS fixed to 15 resources each but varied the number of resources (and thus consumers) at the OEM site from 15 to 45. The first plot shows the "Mean Award Rate at the OEM" metric, which is the probability for a consumer at the OEM site to successfully bid for an input resource. The second plot shows the "Standard Deviation of the Group Size at an RMS" metric, which captures the normalized variation of the number of bids coming in to a RMS site. We applied both metrics in the MiniRAG experiments already and, just as in the MiniRAG implementation, we find the characteristic phase structure of the "underloaded" (small N), the "limited resource" (N near RMS capacity), and the "overloaded" (large N) regime of the distributed resource allocation. Thus, we confirm the correctness of our LogRAG implementation.

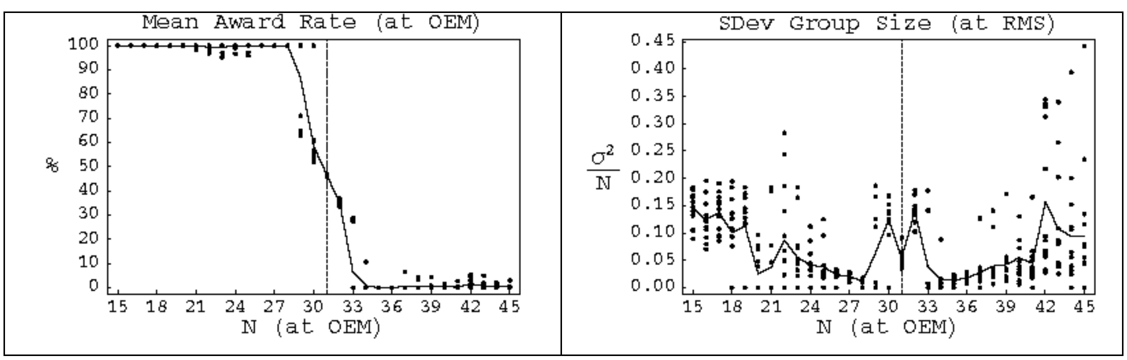

Figure 18. Parameter sweep of MiniRAG configurations.

Next, we explored the dynamics of a larger LogRAG, diagrammed in Figure 19. The nodes are configured at each level to have abundant supply (thus, working up from the bottom level, 2 x 8 > 15, 2 x 15 > 27, 2 x 27 > 49), except that the total supply available to Tier02a is swept from 6 (undersupply relative to Tier02a's demand of 15) to 26 (oversupply).

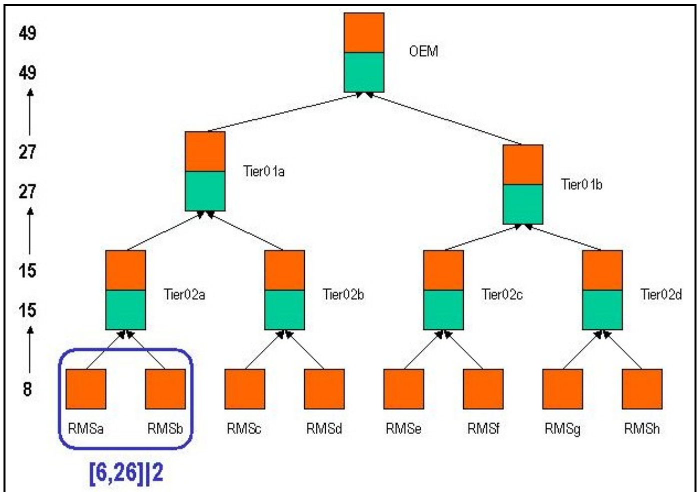

Figure 19: LogRAG test for Disturbance Propagation.—The numbers at the left indicate the supply available (at orange squares) or the demand (at green squares). The total supply available at RMSa and RMSb is swept from 6 to 26 in steps of 2.

At most nodes, the mean award rate as a function of the supply to Tier02a is flat at 100%, as would be expected. However, Figure 20 shows that Tier02a's dependents depend on its level of supply in ways that are not immediately intuitive. On the left, the transition at Tier02a is just what we would expect for an isolated RAG. (The reverse from left to right relative to Figure 18 is because here we are sweeping supply, and there we are sweeping demand.) In the center, one level above Tier02a, the mean award rate is actually higher in the undersupplied region than at the transition point, where it dips sharply. The right-hand figure shows that the OEM also experiences a dip in mean award rate, this time displaced to the left (toward greater undersupply). This dip is a phenomenon of great potential importance in logistics applications, and we look forward to exploring it in more detail in later research.

5 Transition Efforts

AORIST technology has proven valuable in Altarum deployment projects for the military in two domains (supply network engineering and modeling personnel issues), and is instrumental in a new modeling initiative under the Army's Training and Doctrine Command (TRADOC) Analysis Center (TRAC) Monterey. The LogRAG concepts are currently being briefed to potential users in the logistics community. In addition, several publications have appeared, making the results of our research available to others.

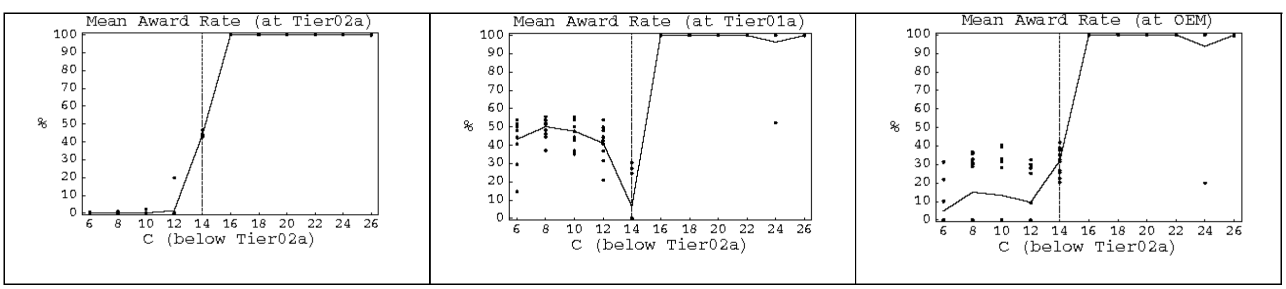

Figure 20: Dependence of Mean Award Rate on supply to Tier02a.—Measured at Tier02a (left), Tier02a (center), OEM (right). Dots are individual experiments; lines are plotted through means.

5.1 Supply Network Engineering

Altarum (and its predecessor organizations ERIM and the Center for Electronic Commerce from ITI) has an active practice with industrial and government customers in engineering supply networks, with particular emphasis on managing their dynamical behavior. Major channels for technology transition from AORIST are through two DoD funded consortia in which we are active: ONR's Supply-chain Practices for Affordable Naval Systems (SPANS), and DLA's Defense Sustainment Consortium (DSC). Altarum is leading technical projects under both programs that will draw on tools and techniques from AORIST. The SPANS Supply Chain Dynamics project focuses on supply chains supporting NorthrupGrumman Newport News, the main shipyard supporting the US nuclear carrier fleet. DSC's Robust Lean Supply Chain project will focus on manufacturing systems at Raytheon.

- The dependence of computational effort on load and capacity is again a central issue in determining effective capacity and balancing leanness against the need for agility. These models are critical in building business cases for capacity that would be reckoned "excess capacity" under traditional cost-accounting models, but that are actually necessary for dynamic stability.
- The impact of Activation Level (AL) on Time To Solution (TTS) in the Color RAG is directly relevant to assessing optimal timing of releases in a supply chain (shipment authorizations sent from customers to their suppliers). In addition, it is likely that the dynamics of convergence depends sensitively on the topology of the supply network, and our methods may be critical in assessing the right degree of fan-out, the impact of lower-tier suppliers serving multiple mid-tier companies that converge again at the first tier, and related structural design issues.
- Our dynamic metrics will be critical for providing decision support for rough-cut capacity planning.
- Adaptive Parameter Search Environment (APSE) can help identify "tight spots" that require more attention (e.g., improved processes, back-up stores, restructuring). In fact, it is already being used in the SPANS project.

5.2 Personnel Models

Altarum is constructing an agent-based model of the Navy's personnel system in support of the Comprehensive, Optimal Manpower & Personnel Analytic Support System (COMPASS) being constructed for the Navy Personnel Research, Studies and Technology Department (NPRST). A critical objective for this model is identifying sets of personnel policies that produce desirable emergent properties for the system as a whole. We are using parameter search methods based on APSE to guide this search.

5.3 TRAC Monterey

Altarum is conducting a research project for TRAC Monterey to evaluate the applicability of multi-agent models for exploring the dependency of C4ISR systems on their environment. We expect to use APSE as a key tool for exploring the dynamic behavior of these systems over their parameter space.

5.4 Logistics Opportunities

Altarum's Supply Chain Engineering practice area has extensive engagements with the logistics and supply community, led by BGEN Robert Mansfield, USAF (ret.), former Special Assistant for Supply Chain Integration and Logistics Transformation, Deputy Chief of Staff for Installations and Logistics, Headquarters U.S. Air Force, Washington, D.C. Techniques developed in AORIST, including APSE and the LogRAG model, are currently being reviewed with our customers in this area, and are generating promising interest.

5.5 Publications

In addition to publications cited in the work described above, several publications have appeared based on work performed prior to the extension. These include a discussion of our extensions of the minority game to support analysis of resource allocation problems [11], an analysis of different forms of agent interaction [7], a discussion of APSE[2] (which received two best-paper nominations at AAMAS 2003), and an analysis of phase changes in the Color RAG [1]. A discussion of the applicability of the concept of universality to multi-agent systems has been accepted for presentation at AAMAS 2004 [9].

Acronyms

AAMAS	International Joint Conference on Autonomous Agents and Multi-Agent Systems
AL	Activation Level
ANT	Autonomous Negotiating Teams
AORIST	Agents Overcoming Resource-IndependEnt Scaling Threats
AP	Assignment Problem
APSE	Adaptive Parameter Search Environment
BGEN	Brigadier General
C4ISR	Command, Control, Communications, Computation, Intelligence, Surveillance, and Reconnaissance
CAMERA	Coordination and Management Environments for Responsive Agents
COMPASS	Comprehensive, Optimal Manpower & Personnel Analytic Support System
DARPA	Defense Advanced Research Projects Agency
DLA	Defense Logistics Agency
DSC	Defense Sustainment Consortium
ISI	Information Sciences Institute
IXO	Information eXploitation Office
LogRAG	Logistics RAG
MAPLANT	Maintenance Planning ANT
MG	Minority Game
NPRST	Navy Personnel Research, Studies and Technology
OEM	Original Equipment Manufacturer
ONR	Office of Naval Research
QuiRT	Quick Review tool
RAG	Resource Allocation Game
RAPSIDy	Resource Allocation Problem Solver Incorporating Dynamics
RTE	Resource Timeline Environment (element of RAPSIDy)

SNAP	Schedules Negotiated by Agent Planners
SPANS	Supply-chain Practices for Affordable Naval Systems
TRAC	TRADOC Analysis Center
TRADOC	Training and Doctrine Command (US Army)
TTS	Time To Solution
TUM	Trivially Unsolvable Matrix
USAF	United States Air Force
VPR	Value Per Resource required

References

[1]S. Brueckner and H. V. D. Parunak. Information-Driven Phase Changes in Multi-Agent Coordination. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, pages 950-951, 2003.

[2]S. Brueckner and H. V. D. Parunak. Resource-Aware Exploration of Emergent Dynamics of Simulated Systems. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, pages 781-788, 2003.

[3]A. Bugacov, S. Fitzpatrick, G. Karsai, V. Lesser, H. V. D. Parunak, V. Raghavan, B. Selman, and P. Szekely. Autonomous Negotiating Teams Program eBook. vol. 2004, 2003. Available at http://www.isi.edu/~szekely/antsebook/ebook/.

[4]D. Challet. The Minority Game's Web Page. vol. 2001, Damien Challet, 2000. Available at http://www.unifr.ch/econophysics/minority/.

[5]P. Hall. On representatives of subsets. *J. London Math. Soc.*, 10:26–30, 1935.

[6]H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2:83-97, 1955.

[7]H. V. D. Parunak, S. Brueckner, M. Fleischer, and J. Odell. A Design Taxonomy of Multi-Agent Interactions. In *Proceedings of Agent-Oriented Software Engineering IV*, pages 123-137, Springer, 2003.

[8]H. V. D. Parunak, S. Brueckner, J. Sauter, and R. Savit. Effort Profiles in Multi-Agent Resource Allocation. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS02)*, pages 248-255, 2002.

[9]H. V. D. Parunak, S. Brueckner, and R. Savit. Universality in Multi-Agent Systems. In *Proceedings of Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*, pages (forthcoming), IEEE, 2004.

[10] R. Savit, S. A. Brueckner, H. V. D. Parunak, and J. Sauter. General Structure of Resource Allocation Games. Altarum, Ann Arbor, MI, 2002. Available at www.erim.org/~vparunak/RAGpaper.pdf.

[11] R. Savit, S. A. Brueckner, H. V. D. Parunak, and J. Sauter. Phase Structure of Resource Allocation Games. *Physics Letters A*, 311:359-364, 2002.